Stable Matching-based Mobility Agent Selection in Distributed Mobility Management

Hyeonjae Jeong, Hongrok Choi, and Sangheon Pack, *Senior Member, IEEE*
Department of Electrical Engineering, Korea University, Seoul, Korea (e-mail:{qeqe, ghdfhrooo, shpack}@korea.ac.kr)

***Abstract*—In partially distributed mobility management (P-DMM), location management functions are centralized whereas data forwarding functions are distributed to DMM gateways (DMM-GWs). Since all packets are processed by DMM-GWs, which one is selected as a serving DMM-GW has a significant impact on the performance of mobile nodes (MNs). In this paper, we propose a stable matching-based DMM-GW selection scheme considering the preferences of both DMM-GWs (w.r.t. load balancing) and MNs (w.r.t. location update cost). Evaluation results demonstrate that the proposed scheme generates about 50% location update cost while the load fairness increased to 0.75 in the most unbalanced mobile distributed state.**

***Index Terms*—Distributed mobility management, stable matching, location management, signaling cost, load balancing.**

I. Introduction

In centralized mobility management (CMM) systems, a mobility agent is centralized at the core and plays key roles in location management and data forwarding. On the other hand, distributed mobility management (DMM) flattens a mobile network by migrating the mobility agent closer to mobile nodes (MNs). Owing to the ability to release the load of CMM, DMM is perceived as a promising approach for a rapidly growing mobile network.

In particular, partially DMM (P-DMM) is one of DMM schemes, consists of central mobility database (CMD) and a distributed mobility management gateways (DMM-GW) [1]. CMD is responsible for storing the location information and mobility sessions of MNs. It maintains the information of which DMM-GW is currently serving MN and which DMM-GW is the previous one. DMM-GW works as an anchor and a default gateway. It forwards delivered packet to MN. A managing area of DMM-GWs is defined by geolocation or a set of cells to provide an optimized path over cells [2]. The cell can be managed by multiple DMM-GWs, which are distributed across a network. As stated in [3], splitting the data plane from a central mobility agent provides relief to massive traffic problems while encouraging the mobile data traffic to optimally route and maintaining legacy with existing PMIPv6.

While P-DMM has many merits for managing a huge number of mobile devices, the performance is not so satisfactory if MN's location management method is not well supported. The location management is a process of updating MN's location through signaling between location managers (i.e., CMD, DMM-GW). When MN moves, CMD revises the location of MN and considers selecting a new GW based on a network state. However, the location management consumes significant network resources, including network bandwidth and computing time at network devices in P-DMM [4]. Especially, the frequent DMM-GW registration update for location management of MN generates a high signaling cost and traffic overhead in the network. Therefore, selecting DMM-GW has a big impact on the network performance.

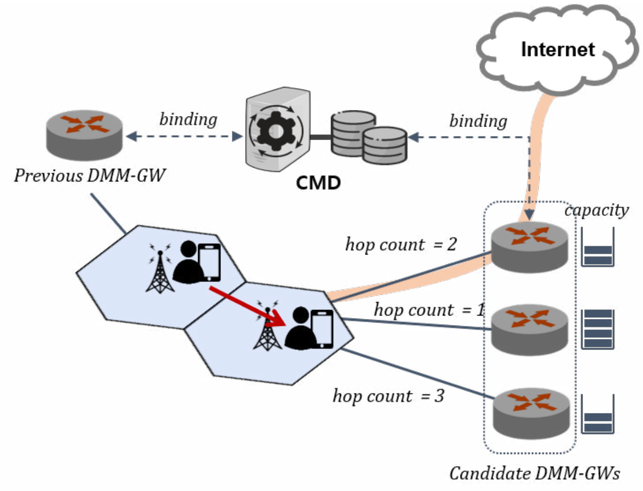
In the literature, several mobility agent selection schemes have been proposed. [5] proposed a mobility agent selection based on a distance between MN and the agent to reduce IP handover latency and packet loss (i.e., distance-based scheme). The nearest mobility agent is selected to reduce registration delay. However, in the distance-based scheme, traffic can be concentrated on some agents which causes load unbalance on the network in case of MNs are crowded in a specific region. [6] forms clusters that comprise several agents. A head agent of its cluster performs intra and inter-cluster communications that provide route optimization and handover latency reduction. [2] proposed handover cost-based mobility anchor selection (CMAS) to select a suitable mobility agent for QoS of MN. However, those researches do not consider the mobility of each device, it does not solve the practical problems of different mobility.

However, a mobility agent selection scheme with one factor cannot bring out an optimal path with low location update cost and load balancing. In the matter of agent selection, it is necessary to select the agent taking into account the requirements of both sides to ensure the performance of both the network and the MN. Therefore, it is important to meet the requirements of both network and MN to improve performances such as load balancing and location update cost reduction. In this paper, we find the optimal combination of mobility agent (same as DMM-GW) and MN through a stable matching algorithm in P-DMM. The simulation results show that the proposed scheme based on the matching algorithm generates about half the location update cost with increased load fairness over the existing scheme even in a highly unbalanced state of distribution of MNs.

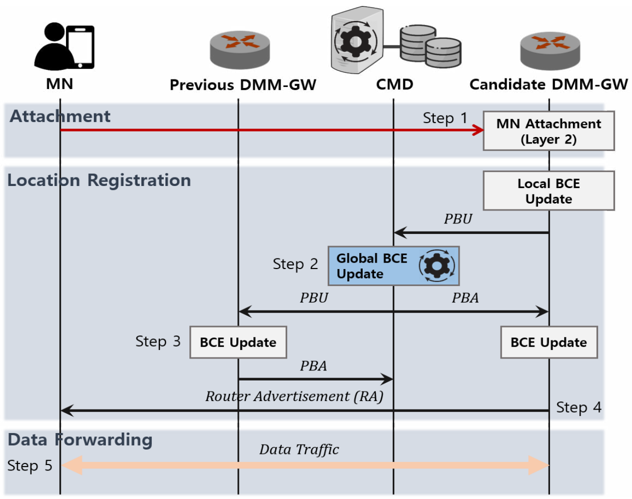
The remainder of this paper is organized as follows. In Section II, the system model is introduced, and the operation of the matching algorithm is described in Section III. Section IV presents evaluation results and Section V concludes this paper.

II. System Model

We consider DMM domain where distributed DMM-GWs serve a large-scale mobile network that consists of MNs with



(a) DMM-GW Selection



(b) DMM-GW registration flow

Fig. 1. Overview of DMM-GW selection and DMM-GW registration flow.

a variety of velocity. MN moves based on its mobility defined by session-to-mobility ratio (SMR) to an adjacent cell to the currently located cell. As shown in Figure 1(a), when the MN moves to an adjacent cell and the serving DMM-GW needs to be changed, one of the candidate DMM-GWs is selected. The nearest DMM-GW is preferentially selected to lower traffic cost. Each DMM-GWs register MNs to be serviced without exceeding their capacity. At the same time, the location change of MN generates DMM-GW registration flow which is a binding signaling consists of proxy binding update (PBU) and proxy binding acknowledgement (PBA) packet between MN, a previous DMM-GW, Candidate DMM-GW and CMD.

Registering MN to a new DMM-GW follows the DMM-GW registration flow [7], the process which causes a signaling cost, as shown in Figure 1(b). We define the candidate DMM-GW that MN accessed sends a PBU message to CMD (first PBU packet in Figure 1(b)). It notifies CMD that MN has changed a location and it needs a new mobility agent. CMD runs a matching algorithm considering the distance between MN and DMM-GW, and mobility of MN. Then CMD processes binding update to associate a new address of MN from a previous DMM-GW to a new DMM-GW in binding cache entry (BCE) (step 2 in Figure 1(b)). The higher the mobility of MNs, the more location update signaling is required, because CMD computes the proper DMM-GW each time MN moves to another cell. Defining the moving event of MN is affected as whether it is out of the area in charge of DMM-GW, but we do not cover the area issue in this paper.

TABLE I
KEY NOTATIONS

Symbol	Meaning
m_i	ith mobile node
g_j	jth DMM-GW
I	a set of m_i
J	a set of g_j
c_j	capacity g_j
μ_i	mobility of m_i
d_{ij}	hop distance between m_i and g_j
$P(m_i)$	preferece list of m_i
$P(g_i)$	preferece list of g_j

Meanwhile, if a large number of nodes are positioned in a specific area, DMM-GW nearby a crowded area can be overloaded. Hence, we adopt zipf's distribution to quantify the unbalanced state of across the cell considering DMM domain. MNs are distributed according to location popularity which is subject to zipf's distribution as

$$P(i) = \frac{i^{-\alpha}}{\sum_{k=1}^{M} k^{-\alpha}}, \quad (1)$$

where α is the zipf's distribution parameter. A smaller α indicates lower location similarity. For instance, if $\alpha = 0$, the location of MNs has a uniform distribution.

III. Stable Matching-based DMM-GW Selection

A stable matching has been studied extensively, beginning with the pioneering work of Gale and Shapley [8]. A stable pair is created by a proposal of one set to the other set and the pair is optimized for a proposed side in the matching algorithm. A matching takes into account the preferences of each element of one set toward the elements of the opposite set and vice versa. Especially, we applied a many-to-one stable matching [9] to pair several MNs to one DMM-GW. Hence, we first specify the preference list of each set.

A. Preference List Construction

We first describe how to construct preference lists for stable matching. Let m_i (i = 1, · · ·, I) be MN, and m_i have a different μ_i which presents the mobility of m_i. Let g_j (j = 1, · · ·, J) be the set of DMM-GWs. c_j represents the capacity of each DMM-GW which is the number of MN registered in g_j. For a given m_i and g_j, let d_{ij} be the wired network hop count between them. $P(m_i)$ is MN's preference list over DMM-GWs. For MN, it is advantageous to register in DMM-GW whose distance is close to receive data in low traffic cost, so the list is made in the order of the closest distance. DMM-GW's preference list over MNs $P(g_j)$, on the other hand, is listed according to the mobility of MN to lower location update frequency. Table I summarizes the key notations for the ease of reference.

B. DMM-GW Selection

The algorithm is described in Algorithm 1. In order to perform the matching algorithm, it is necessary to create preference lists of MNs and DMM-GWs. We defined that CMD has information about the distance between MNs and DMM-GWs, and the mobility information such as the past registered DMM-GW, current location, cell crossing rate of every MN [7]. First, g_j initializes temporary match-list $temp_j$ (line 3 in Algorithm 1). Then both sides set their preference list starting with the most preferred one from set of opponent (line 4 - 9 in Algorithm 1). After that, based on the preference list of MNs, each m_i proposes to P_i^*, the most preferred DMM-GW (line 12 in Algorithm 1). DMM-GW g_j should consider its capacity c_j before accepting the proposal. If the capacity is enough, g_j accepts the proposal. $temp_j$ is updated to reg_j that added the accepted MN and c_j is increased (line 13 - 16 in Algorithm 1).

If the capacity is full, P_j is also considered. g_j selects $temp_j^*$, and compares with the proposed MN's μ_i. $temp_j^*$ is MN that has highest mobility among $temp_j$. If the μ_i of the proposed MN is smaller than $temp_j^*$, $temp_j^*$ is replaced with m_i, and $temp_j$ is updated (line 18 - 21 in Algorithm 1). The algorithm is repeated until all MNs propose to DMM-GW. But if g_j has better preference on $temp_j^*$, m_i is rejected. Rejected MN goes back to step 2 and proposes to next preferred DMM-GW. In an area with heavy load, the probability of rejection of a fast MN increases, so it is registered in a remote DMM-GW which lowers frequent location updates. Eventually, each DMM-GW registers a slow MN first and MN at least avoids and lowers location updates.

If there exists the blocking pair, it means that MN proposed admits less preferred DMM-GW in the iterative or DMM-GW refuses the proposal and its capacity is enough. However, in each round, the most preferred DMM-GW receives the proposal from MN. Also, DMM-GW does not refuse the proposal until its capacity is full. Therefore, the above situation will not happen. Thus, the matching will not cause a blocking pair.

Algorithm 1 DMM-GW Selection Algorithm

1: **Inputs:**
    $m_i \in M$, $g_j \in G$, d_{ij}, μ_i, c_j
2: **Outputs:**
    A perfect matching between M and G
3: **Initialize:**
    $temp_j = \{\}$
4: **for** i = 1 to I **do**
5:     P_i: ascending order of d_{ij}
6: **end for**
7: **for** j = 1 to J **do**
8:     P_j: ascending order of μ_i
9: **end for**
10: **for** i = 1 to I **do**
11:     $reg_j \longleftarrow temp_j$
12:     m_i proposes P_i^* ($g_j \longleftarrow P_i^*$)
13:     **if** $|temp_j| < c_j$ **then**
14:         Add m_i to $temp_j$ and make new reg_j
15:         $c_j = c_j + 1$
16:         $i = i + 1$
17:     **end if**
18:     **if** $|temp_j| = c_j$ **then**
19:         **if** $\mu_i < temp_j^*$ **then**
20:             Replace $temp_j^*$ with m_i and make new reg_j
21:             $m_i \longleftarrow temp_j^*$
22:             Go to line (12)
23:         **else**
24:             Go to line (12)
25:         **end if**
26:     **end if**
27: **end for**

IV. Performance Evaluation

We developed our proposed scheme with 5 DMM-GWs and 1000 MNs by using MATLAB. Each MN moves to neighbor cells according to its mobility rate. The mobility rate, also called a cell crossing rate, follows a gamma distribution [1]. We compare the proposed algorithm, DMM-GW selection based on the matching algorithm, with the random selection and the distance-based selection scheme. The random selection scheme selects DMM-GW randomly and the distance-based selection scheme selects the nearest DMM-GW.

A. Effect of SMR

In order to show the effect of SMR, we fixed a zipf's parameter to 1.0 and the session ratio of SMR to 1 to observe the effect of the actual movement on MNs. The total signaling cost is generated across DMM-GWs according to each of the selected mechanism. All schemes are applied when handover occurs when the MN moves to another cell. That is, the MN with high mobility generates more DMM-GW selection events. As shown in Figure 2, as SMR increases, the total signaling cost gets lower at three schemes. This is because the high mean mobility of MNs causes frequent handover and location update, which leads to generate the high signaling cost. On the other hand, handover occurs much less because MNs mainly stay in the current location. Therefore the cost gets lower when the mean movement speed is low.

Also, it can clearly be seen that the proposed scheme outperforms other schemes by having 54% cost reduction compared to the random selection scheme and 29% reduction compared to the distance-based scheme. The reason is that the matching scheme considers not only the distance between MN and DMM-GW, but also the user's mobility. With the distance-based scheme, a fast-moving MN creates location update to DMM-GWs often. The matching scheme allocates MN with high mobility to a far DMM-GW so that it can lower frequent location update of MN.

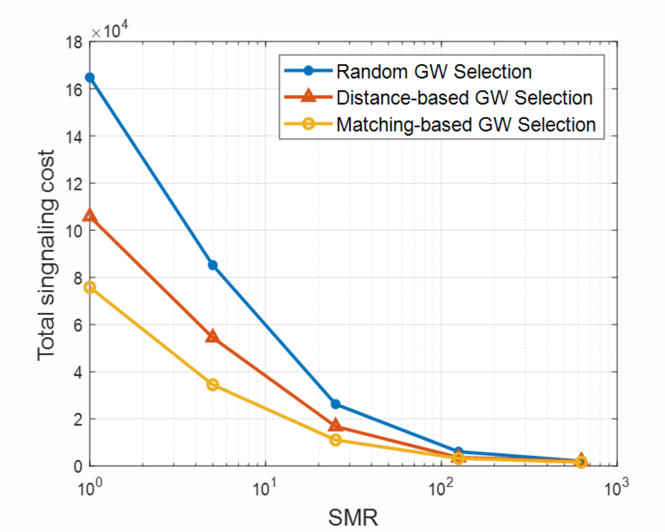


Fig. 2. Effect of SMR to the signaling cost.

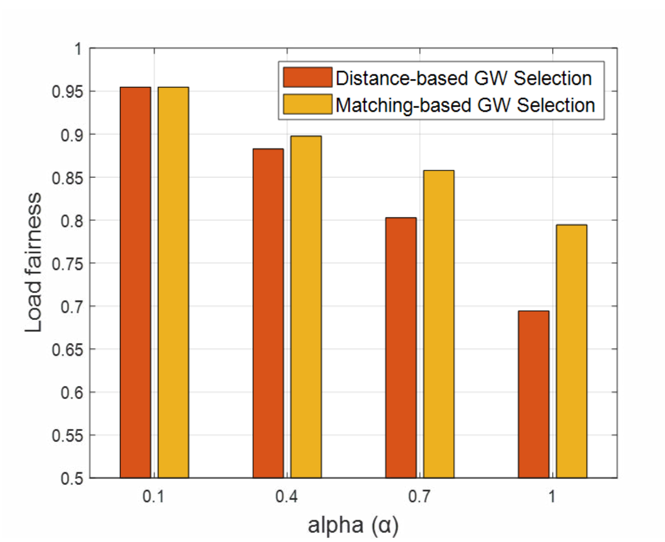


Fig. 3. Effect of MN's location distribution to the load fairness.

B. Effect of α

We use different scenarios in terms of location similarity, which is α, to show the load fairness of each scheme according to the distribution of MNs. We fix SMR of MN to 1, and the zipf's parameter α was changed by 0.1 to 1.0. In this simulation, we consider the load balance in the operator's network with Jain's load fairness index [10] to prove that our proposed scheme has better load equality. The Jain's fairness index is followed as

$$F(L) = \frac{(\sum_{g=1}^{G} L_j)^2}{G \sum_{g=1}^{G} {L_j}^2} \tag{2}$$

while L_j is the load of the g_j. If the $F(L)$ value is 1, it means perfect fairness which implies that all DMM-GWs have same load.

Figure 3 shows the load fairness $F(L)$ for different values of α. We compared our proposed scheme with the distance-based DMM-GW selection scheme. Overall, both schemes showed that load fairness decreased as the distribution of MNs became concentrated. This is because all MNs cannot be allocated to the closest DMM-GWs due to the capacity limitation of DMM-GW when the local density increases (i.e., alpha increases) in both schemes. However, the matching algorithms scheme shows better load fairness than that of the distance-based scheme. Since all MNs cannot be allocated to the closest DMM-GWs due to the capacity limitation, some MNs should be allocated to other DMM-GWs and thus a load of DMM-GWs can be distributed. Especially, the rejected MN is assigned to the furthest DMM-GW, so the local load is relieved and the overall traffic is well distributed widely.

V. Conclusion

In this paper, we propose a DMM-GW selection scheme based on matching theory in P-DMM domain. By considering conflicting requirements of MN and the network, we can select optimal DMM-GW through a stable matching. The evaluation shows our proposal has better performance in terms of DMM-GW registration signaling cost and load balancing than existing studies. In future work, we plan to extend the proposed algorithm to implement in a more practical environment.

Acknowledgement

This work has been supported by the Future Combat System Network Technology Research Center program of Defense Acquisition Program Administration and Agency for Defense Development (UD190033ED).

References

[1] F. Giust, "Distributed mobility management for a flat architecture in 5G mobile networks: Solutions, analysis and experimental validation," Doctoral dissertation, Universidad Carlos III de Madrid, Spain, pp. 1–195, March 2015.
[2] B. Davaasambuu, T. Telmuun, D. Sasko, Y. Keping, and S. Sodbileg, "A novel anchor selection scheme for distributed mobility management", *Computer Science*, vol. 22, no. 1, February 2021.
[3] J. C. Zuniga, C. J. Bernardos, A. de la Oliva, T. Melia, R. Costa, and A. Reznik, "Distributed mobility management: A standards landscape," *IEEE Communications Magazine*, vol. 51, no. 3, pp. 80-87, March 2013.
[4] D. Shin, D. Moses, M. Venkatachalam, and S. Bagchi, "Distributed mobility management for efficient video delivery over all-IP mobile networks: Competing approaches," *IEEE Network*, vol. 27, no. 2, pp. 28-33, March-April 2013.
[5] Y. Xu, H. C. J. Lee, and V. L. L. Thing, "A local mobility agent selection algorithm for mobile networks," in *Proc. IEEE ICC*, May 2003.
[6] S. M. Ghaleb, S. Subramaniam, Z. A. Zukarnain, and A. Muhammed, "Load balancing mechanism for clustered PMIPv6 protocol," *EURASIP Journal on Wireless Communications and Networking*, vol. 135, May 2018.
[7] M. K. Murtadha, N. K. Noordin, B. M. Ali, and F. Hashim, "Design and evaluation of distributed and dynamic mobility management approach based on PMIPv6 and MIH protocols," *Wireless Networks*, vol. 21, no. 8, pp. 2747–2763, November 2015.
[8] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, January 1962.
[9] J. Sethuraman, C. P. Teo, and L. Qian, "Many-to-one stable matching: Geometry and fairness," *Mathematics of Operations Research*, vol. 31, no. 3, pp. 581-596, 2006.
[10] R. K. Jain, D. M. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA, September 1984.